

DATABASE Expert

Various torments are afflicting readers this month, as they struggle with unopenable emails, dates in a parameter query and values in a table. Fortunately, **Kay Ewbank** is on hand to fight those database demons



NEW IMPROVED WORKBENCH

The Access Workbench, a tool designed to help you switch between Access versions and different databases to carry out database maintenance and design, has been updated.

New features in version 4.4.6 include MDE compiling, the ability to change database startup options without opening the database and better support for decompiling. The Workbench allows you to switch between Access 97, 2000, 2002 and 2003 versions, and compact and back up databases.

The Workbench costs £51 including VAT and is available from www.vb123.com/workbench.

EASY ACCESS TO SQL

A spreadsheet-style view of SQL Server 2005 data is the promise behind Krell Software's OmniView. This allows viewing, editing, sorting and grouping of SQL Server data as though you were working in a spreadsheet.

The new version costs £28 including VAT and has been updated to support SQL Server 2005. The idea behind the software is to make access to database information easy for a wide range of users regardless of their skill levels.

For more information, visit www.krell-software.com/omniview.

TO SQL AND BACK AGAIN

DMSoft Technologies has released a new data converter that can convert and synchronise databases between Microsoft Access and the open-source

PostgreSQL. The converter supports two-way conversion, so you can move back to Access if you need to, having initially converted to PostgreSQL. You also get the option of converting specific objects in a database.

Access2PostgreSQL Sync costs £31 inc VAT from <http://data-conversions.net/access-to-postgresql-sync-converter.html>.



Q We have a Microsoft Access database and want to use the values in one table to control the values in a second. We want to ensure that when a user enters data into a field in table one (or into a form based on table one), the value entered doesn't appear in table two. How do I set up a validation rule to check this and where do I enter it?

Bob Laithwaite

A What you're trying to do is a little strange. The best way to ensure that values aren't duplicated is to keep data of the same type in the same table. I can imagine a situation when you might want to store temporary data in a separate table. Maybe you have a stock control system with a main products table and also a separate temporary products table so you can record items that you stock temporarily. Even in a situation such as this, though, it is better to keep all the products in the same table and have an extra column indicating whether they are temporary or not. Think carefully about why you need to carry out a cross-table referential check; it could mean you have made a mistake in your database design.

It may also be tricky to get a definitive answer as to whether or not there is a clash between records. If we consider our two product tables, how can you be sure that the teapot you are entering as a new record in one table is the same as the teapot already entered in the other table? The only way to be sure is to check all the fields, or compare the product id code from the product table. If you need to check the product id code, this also suggests that the data you're entering belongs in the product table.

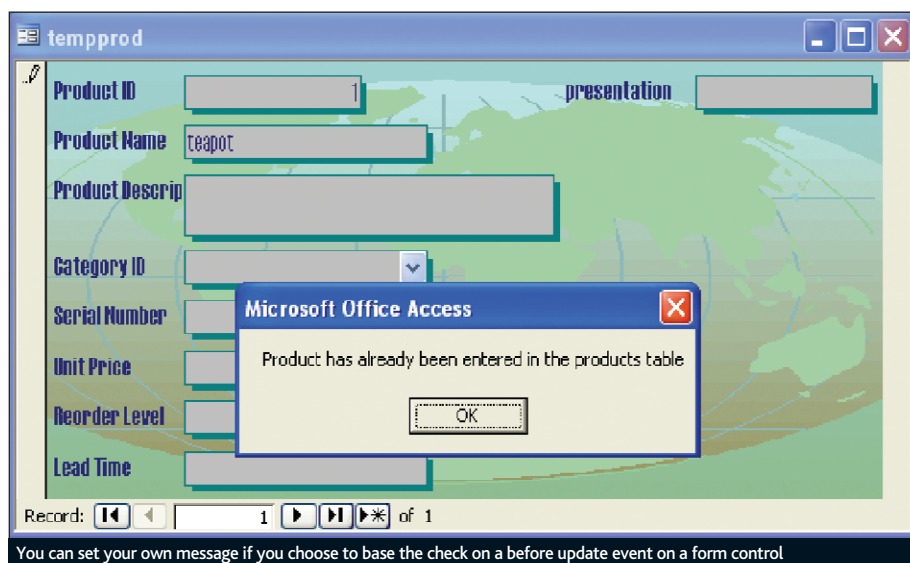
Assuming you want to press on with your current database design, there are a couple of

things you can do. Always use a form to enter the data, and use the Control_BeforeUpdate event to check that the value being entered doesn't already exist in the other table. For example, suppose the form is used to enter temporary products in our mythical stock control system. If you're entering a product name, you could attach this code to the product name field on your form:

```
Private Sub ProductName_BeforeUpdate
    (Cancel As Integer)
If (Not IsNull(DLookup("[ProductName]", "Products", "[ProductName] = '" & Me!ProductName & "'"))) Then
    MsgBox "Product has already been entered in the products table"
    Cancel = True
    Me!ProductName.Undo
End If
End Sub
```

This uses the VBA DLookup function to check for matching values in the main product table. If it finds a match, it displays a message and undoes your edit of the ProductName field.

There is another way to achieve what you want using a well-hidden feature of Access. In your code, you can include SQL statements to carry out actions. This isn't usually necessary, but there is a facility that seems only accessible in this way. SQL provides a CHECK clause that can be used to constrain the values entered into a column in a table. To make this work, you need to close your database and open another where you set up a temporary form on which you put a command button. Set its code along the following lines:





ACCESS ATTACHMENT WON'T OPEN

Q I emailed a friend an Access database that I created. When she tried to open it, a message announced that it came from an 'untrusted' source and could not be opened. I then emailed the same file to myself and I can't open it either. How do I overcome this?

Eric Bird

A It's irritating to be told that your own database comes from an untrusted source. However, the real villain of the piece here is not Microsoft but the hackers who force software companies to put seemingly draconian security in place.

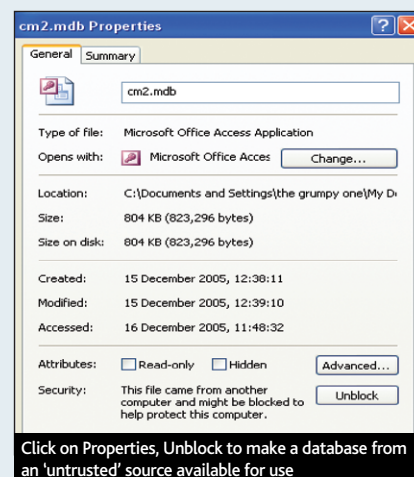
If you are using Windows XP with Service Pack 2 installed, you will get this message. According to the Microsoft documentation, even if you compress the file using PKZip or a similar utility, Windows will recognise the file

as being a potential threat to your system and will continue to block it.

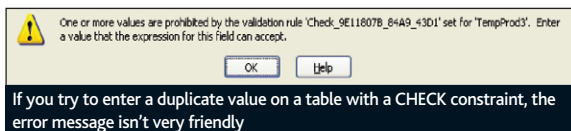
We tried emailing various files and, like you, we found we weren't allowed to save or open an MDB file sent as a Microsoft Outlook attachment. But if the same file was zipped first, we could save the compressed attachment to our hard disk, unzip it and open it.

Another workaround is to rename the file before sending it, adding an extension that isn't on the blocked extensions list. For instance, if you wanted to send the file *shopper.mdb*, you could rename it as *shopper.mdb_extra*. This would be passed through without problems, though the recipient would have to rename it as *shopper.mdb* before they could use it in Access.

If you've downloaded a file that is blocked, you can unblock it. Open Windows Explorer from All Programs, Accessories in the Start menu. Navigate to the folder where the file is located.



Right-click on the blocked database file. View Properties and click on the button labelled Unblock at the lower right of the general tab. The file should then open.



```
Private Sub Command0_Click()
Dim ADOconnection As New ADOBD.
Connection
Dim strSQL As String
strSQL = "CREATE TABLE TempProd ("
strSQL = strSQL & "ProdName VARCHAR
(50), ProdType VARCHAR(20), "
strSQL = strSQL & "CHECK (ProdName
NOT IN (SELECT ProductName FROM
Products)))";
With ADOconnection
.Provider = "Microsoft.Jet.OLEDB.
4.0"
.Open "Data Source=c:\data\
shopper.mdb"
.Execute (strSQL)
End With
End Sub
```

This form will be used only once to create a table TempProd in the Shopper database. You can close the form and database and re-open the Shopper database, where you'll see a new table called TempProd.

The new table is a bit minimal, as we've defined just two columns: prodname and prodtype. In a real database you'd need more, but you could add extra columns in the normal fashion after the table has been created using the code above. The clever part of the code is the part that says:

```
CHECK (ProdName NOT IN (SELECT
ProductName FROM Products)))
```

This tells Access (via SQL) that the table has a built-in rule that says a product name entered in the TempProd table cannot already exist in the ProductName column in the Products table.

To make this code work, you'll need to add references to the most recent Microsoft ActiveX Data Objects Library and the most recent Microsoft

ADO Ext. DDL and Security. Pick those with the biggest number from the list that appears when you choose Tools, References.

If you try to enter a value in TempProd that is already entered elsewhere, you'll get an error message informing you that the value you've entered is prohibited by the validation rule.

Q I am using a parameter query to display all records when the date is in a specific week. At present, the user has to enter the start date and end date. As it is always a week, is there a way that I can ask for just the start date and add six days to it for the end date? In the same vein, can I capture the entered parameter and save it as a variable to use in VBA or report headings?

Ian McArthur

A You can certainly use the start date and add six days to it. If your parameter query has a parameter called [Start Date] with a data type of time/date, you can use a criterion for selection such as:

```
>=[Start Date] And <=[Start Date]+6
```

You can also show the end date in a report based on your parameter query. Just add a text box to the report header and set its control source to:

```
= "Report from " & [Start Date] & " to " & [Start Date]+6
```

You can change the text shown to suit your purposes. When someone runs this report, they'll be asked for the start date before the report is created.

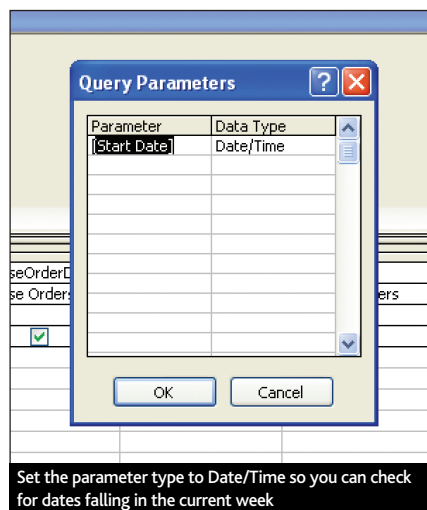
Bear in mind that if the parameter query returns no data, the parameter will not appear in the header of the empty report. Instead, you'll see an #Error message, which might worry your users.

To get around this, you have to change the way your users enter the parameter. If you create a form with a text box and use that as the source for the criterion for the query, the information on the dates selected will appear, whether or not any data is returned. To use a form control as a query parameter, refer to it as:

```
[Forms]![myform]![Start Date]
```

Here myform is the form on which the user entered the Start Date value.

The ease with which you can use the parameter in Visual Basic for Applications (VBA) depends on where you are running VBA and where the query parameter was entered. The simplest method is to enter the query parameter on a form as suggested above; you can then refer to the form control in your VBA code. **ES**



NEXT MONTH

SOFTWARE HELP

Next month database coverage in *Computer Shopper* moves to our new *Software Help* section. This will deal with queries relating to all office and creative applications. Send your questions to:

softwarehelp@computershopper.co.uk